Amendments to the Specification

Please replace the paragraph beginning at page 4, line 12, with the following rewritten paragraph:

Figure 6 is a diagram of an equivalent processing matrix for computing the modulated discrete cosine transform and its inverse together within the mixed lossless compression process of Figure 5.

Please replace the paragraph beginning at page 4, line 19, with the following rewritten paragraph:

Figure 9 is a graph showing reference samples used for a multi-channel least means square predictive filter in the pure lossless compression of Figure 7.

Please replace the paragraph beginning at page 9, line 13, with the following rewritten paragraph:

For multi-channel audio data, the multiple channels of noise-shaped frequency coefficient data produced by the weighter (142) often correlate. To exploit this correlation, the multi-channel transformer (150) can apply a multi-channel transform to the audio data of a tile. In some implementations, the multi-channel transformer (150) selectively and flexibly applies the multi-channel transform to some but not all of the channels and/or critical bands in the tile. This gives the multi-channel transformer (150) more precise control over application of the transform to relatively correlated parts of the tile. To reduce computational complexity, the multi-channel transformer (150) uses a hierarchical transform rather than a one-level transform. To reduce the bit rate associated with the transform matrix, the multi-channel transformer (150) selectively uses pre-defined (e.g., identity/no transform, Hadamard, DCT Type II) matrices or custom matrices, and applies efficient compression to the custom matrices. Finally, since the multi-channel transform is downstream from the weighter (142), the perceptibility of noise (e.g., due to subsequent quantization) that leaks between channels after the inverse multi-channel transform in the decoder (200) is controlled by inverse weighting. For additional detail about multi-channel transforms in some embodiments, see the section entitled "Flexible Multi-Channel

Transforms" in the related application entitled, "Architecture And Techniques For Audio Encoding And Decoding." Alternatively, the encoder (100) uses other forms of multi-channel transforms or no transforms at all. The multi-channel transformer (150) produces side information to the MUX (190) indicating, for example, the multi-channel transforms used and multi-channel transformed parts of tiles.

Please replace the paragraph beginning at page 10, line 3, with the following rewritten paragraph:

The quantizer (160) quantizes the output of the multi-channel transformer (150), producing quantized coefficient data to the entropy encoder (170) and side information including quantization step sizes to the MUX (190). Quantization introduces irreversible loss of information, but also allows the encoder (100) to regulate the quality and bit rate of the output bit stream (195) in conjunction with the controller (180). The quantizer can be an adaptive, uniform, scalar quantizer that computes a quantization factor per tile and can also compute per-channel quantization step modifiers per channel in a given tile. The tile quantization factor can change from one iteration of a quantization loop to the next to affect the bit rate of the entropy encoder (170) output, and the per-channel quantization step modifiers can be used to balance reconstruction quality between channels. In alternative embodiments, the quantizer is a non-uniform quantizer, a vector quantizer, and/or a non-adaptive quantizer, or uses a different form of adaptive, uniform, scalar quantization.

Please replace the paragraph beginning at page 12, line 23, with the following rewritten paragraph:

The inverse multi-channel transformer (240) receives the entropy decoded quantized frequency coefficient data from the entropy decoder(s) (220) as well as tile pattern information from the tile configuration decoder (230) and side information from the DEMUX (210) indicating, for example, the multi-channel transform used and transformed parts of tiles. Using this information, the inverse multi-channel transformer (240) decompresses the transform matrix as necessary, and selectively and flexibly applies one or more inverse multi-channel transforms to the audio data of a tile. The placement of the inverse multi-channel transformer (240) relative to the inverse quantizer/weighter (250) helps shape quantization noise that may leak across

channels due to the quantization of multi-channel transformed data in the encoder (100). For additional detail about inverse multi-channel transforms in some embodiments, see the section entitled "Flexible Multi-Channel Transforms" in the related application entitled, "Architecture And Techniques For Audio Encoding And Decoding."

Please replace the paragraph beginning at page 13, line 14, with the following rewritten paragraph:

The inverse frequency transformer (260) receives the frequency coefficient data output by the inverse quantizer/weighter (250) as well as side information from the DEMUX (210) and tile pattern information from the tile configuration decoder (230). The inverse frequency transformer (260) applies the inverse of the frequency transform used in the encoder and outputs blocks to the overlapper/adder (270).

Please replace the paragraph beginning at page 13, line 19, with the following rewritten paragraph:

The overlapper/adder (270) generally corresponds to the partitioner/tile configurer (120) in the encoder (100). In addition to receiving tile pattern information from the tile configuration decoder (230), the overlapper/adder (270) receives decoded information from the inverse frequency transformer (260) and/or mixed/pure lossless decoder (222). In some embodiments, information received from the inverse frequency transformer (260) and some information from the mixed/pure lossless decoder (222) is pseudo-time domain information – it is generally organized by time, but has been windowed and derived from overlapping blocks. Other information received from the mixed/pure lossless decoder (222) (e.g., information encoded with pure lossless coding) is time domain information. The overlapper/adder (270) overlaps and adds audio data as necessary and interleaves frames or other sequences of audio data encoded with different modes. Additional detail about overlapping, adding, and interleaving mixed or pure losslessly coded frames are described in following sections. Alternatively, the decoder (200) uses other techniques for overlapping, adding, and interleaving frames.

Please replace the paragraph beginning at page 23, line 1, with the following rewritten paragraph:

Figure 11 depicts the windowing functions applied to original PCM frames of the input audio signal to produce the windowed coding frames for lossy, mixed lossless and pure lossless coding. In this example, the encoder's user has designated a subsequence 1110 of the original PCM frames of the input audio signal 1100 as lossless frames to be encoded with pure lossless coding. As discussed in connection with Figure 5, lossy coding in the presently described unified lossy and lossless compression embodiment applies a sin window 1130 to the current and previous PCM frames to produce the windowed lossy coding frame 1132 that is input to the lossy encoder. The mixed lossless coding of isolated mixed lossless coding frame 1136 also uses the sin-shape window 1135. On the other hand, the pure lossless coder uses a rectangular windowing function 1140. The mixed lossless coding for transition between lossy and lossless coding (at first and last frames of the subsequence 1110 designated for pure lossless coding) effectively combines the sine and rectangular windowing functions into first/last transition windows 1151, 1152 to provide transition coding frames 1153, 1154 for mixed lossless coding, which bracket the pure lossless coding frames 1158. Thus, for the subsequence 1110 of frames (numbered s through e) designated by the user for lossless coding, the unified lossy and lossless compression embodiment encodes frames (s through e-1) using lossless coding, and frame e as mixed lossless. Such a windowing functions design guarantees that each frame has the property of archiving critical sampling, meaning no redundant information is encoded and no sample is lost when the encoder changes among lossy, mixed lossless, and pure lossless frames. Therefore, seamlessly unifying lossy and lossless encoding of an audio signal is realized.

Please replace the paragraph beginning at page 25, line 20, with the following rewritten paragraph:

The storage (1440) may be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, CD-RWs, DVDs, or any other medium which can be used to store information and which can be accessed within the computing environment (1400). The storage (1440) stores instructions for the software (1480) implementing the audio encoder that generates and compresses quantization matrices.